

Pelihakmon 3D-mallintaminen sculptaus tekniikalla

Katsaus toimivan 3D-mallin luomisesta pelimoottoria varten



Ammattikorkeakoulututkinnon opinnäytetyö

Tietotekniikan koulutusohjelma, Riihimäki

Kevätlukukausi, 2018

Simo Happonen

Tietotekniikan koulutusohjelma
Riihimäki

Tekijä	Simo Happonen	Vuosi 2018
Työn nimi	Pelihahmon 3D-mallintaminen sculptaus tekniikalla	

TIIVISTELMÄ

Opinnäytetyöni tarkoituksena on perehtyä pelihahmojen 3D-mallinnusprosessiin. Työ syvennyy 3D-mallinnuksen taustalla piilevän teorian lisäksi, erityisesti viime vuosien aikana yleistyneeseen sculptaus tekniikkaan ja sitä hyödyntävien ohjelmistojen mahdollisuuksiin, erityisesti pelialalla.

Käytännön osuudessa syvennyttään varsinaiseen mallin luomisprosessiin ja tutkitaan asteittain eri työvaiheita, niissä huomioitavia asioita ja miten ne vaikuttavat haluttuun lopputulokseen. Varsinaisen mallinnusprosessin tutkimisen lisäksi, olennaisina osina työtä on mallin re-topology käsittely: hahmon polygonimäärän vähentäminen turhan kuormituksen vähentämiseksi, sekä mallin teksturointi: haluttujen tekstuurien appliointi 3D-malliin optimaalisimmalla tavalla, luoden mahdollisimman fotorealistisen lopputuloksen.

Opinnäytetyön tavoitteena on luoda katsaus mitä vaaditaan toimivan, sekä teksturoidun 3D-mallin luomiseen erityisesti sculptaus tekniikkaa hyödyntäen ja pelimoottorikäyttöä varten, ja kuvata työprosessia mallintajan kannalta.

Avainsanat 3D-mallinnus, peligrafiikka, pelimoottori

Sivut 35 sivua

Information Technology
Riihimäki

Author	Simo Happonen	Year 2018
Subject	Creating a 3D-model utilizing sculpting software	

ABSTRACT

The goal of this project was to create an understanding of what is needed to create a working 3D-model for game-engine use utilizing sculpting software, and to visualize what this process is like from the viewpoint of a 3D-artist.

Along with familiarizing the reader with the software used for this project, the practical part of this thesis is focused on highlighting the step by step process of sculpting a 3D-model, creating and applying the textures and importing the model into the game-engine, as well as what should be kept in mind during the modeling project. Other essential parts of the project include the re-topology work of the model: re-arranging the 3D-model topology to minimize the load on the used hardware to handle the model, and the optimal way to create and apply the desired textures, to reach the most photorealistic result as possible.

The purpose of this project was to study the process of 3D-modeling a character to be used in a video game. Along with getting acquainted with the theory behind 3D-modeling and other related subjects, this project was strongly based on digital sculpting, the digital sculpting software and the possibilities these hold, concerning the game industry.

Keywords 3D-modeling, game-graphics, game-engine

Pages 35 pages

SISÄLLYS

1	JOHDANTO.....	1
2	TEORIA	2
2.1	3D-Grafiikka / 3D-mallinnus	2
2.2	Digitaalinen Sculptaus	3
2.3	Re-topology	4
2.4	Pelimoottori	5
2.5	Teksturointi / UV-mapping.....	5
3	OHJELMISTOT	7
3.1	Mallinnusohjelmisto / Sculptris	7
3.2	Pelimoottori / Unity	8
4	KÄYTÄNNÖN PROJEKTI	10
4.1	Mallin sculptaus	10
4.1.1	Muodon hahmottelu	11
4.1.2	Mallin yksityiskohdat	14
4.2	Re-topology	17
4.2.1	Suunnittelu ja tavoitteet	17
4.2.2	Käytännön käsittely	18
4.3	Teksturointi	21
4.3.1	Tekstuurikartan luonti ja maalaus.....	22
4.3.2	Tekstuurikuvien luonti ja implementointi	25
4.3.3	Tekstuurikartan hienosäätö	28
4.4	Tuominen pelimoottoriin	30
5	POHDINTAA	32
	LÄHTEET	34

1 JOHDANTO

Digitaalisen sculptaus tekniikan kehittyessä, yhä fotorealistisempien hahmojen 3D-mallinnus on tullut mahdolliseksi ja helpommaksi sculptaus-ohjelmistojen vapaamuotoisen mallinnustyylin johdosta. Tämän johdosta peli- sekä elokuvateollisuudessa käytettyjen grafiikkojen ja tuotetun CGI:n taso on noussut räjähdysmäisesti viime vuosikymmenen aikana.

Tämän opinnäytetyön tavoitteena on mallintaa 3D-malli erityisesti videopelikäyttöä varten, ja sculptaus tekniikkaa hyödyntäen, luoden ymmärrystä lukijalle mitä työprosessi pitää sisällään. Lopputulos jota tässä työssä halutaan saavuttaa, on mahdollisimman fotorealistinen ja toiminnallinen hahmo, jota voisi käyttää videopelissä pelimootorin kautta. Tämä lopputulos koetetaan saavuttaa mallintamalla hahmo digitaalisella sculptaus ohjelmalla, suorittaa sille re-topology jälkikäsittely sen teknisen käytettävyyden optimoimiseksi, ja luoda sille mahdollisimman luontevat ja realistiset tekstuurit sen viimeistelyksi, kiinnittäen huomiota mahdollisiin prosessin ongelmatekijöihin ja kuinka niitä tulisi välttää ja mahdollisesti ratkoa.

Tarkoituksena tässä projektissa on simuloida ja hahmottaa minkälainen nykypelihahmojen tuotantoprosessi on 3D-artsilta varsinaiselle pelikehittäjälle, ja mitä kaikkea tämä kyseinen tuotantoprosessi pitää sisällään.

2 TEORIA

2.1 3D-Grafiikka / 3D-mallinnus

Termillä 3D-grafiikka tarkoitetaan grafiikkaa jolla kuvataan kolmiulotteista representaatiota geometrisestä datasta, mikä on yleisesti kuvattuna Cartesiaalisessa koordinaatistossa. (Franklin, 2000).

3D-mallinnus on yhdistelmä matematiikkaa, geometriaa ja suunnittelutyötä. 3D-mallinnuksella yleisesti tarkoitetaan geometrisen datan luomisesta kuvailemaan valmista objektia tai mallia. Jos pohjapiirustus on kaksiulotteinen kuvaus rakennuksesta, 3D-malli samasta rakennuksesta on matemaattinen kuvaus rakennuksen pinnasta kolmiulotteisessa maailmassa. Ohjelmistot joita käytetään 3D-mallintamiseen, hyödyntävät useita erilaisia tekniikoita kuvaamaan objekteja kolmiulotteisessa tilassa. (Wodehouse, 2016).

3D-malleja voidaan kuvata usealla tavalla kolmiulotteisessa ympäristössä mutta yleisimmät ovat:

Polygonaalinen mallinnustapa perustuu pisteisiin joita kutsutaan nimellä "vertex". Vertex pisteet yhdistetään toisiinsa suorilla jotka muodostavat ns. polygonaalisen verkoston jota kutsutaan nimellä "mesh". Suuri osa 3D-malleista mallinnetaan juuri tällä tavalla. (Ghosh, 2014).

Curve-mallinnus eroaa polygonaalisesta mallinnuksesta siinä, että mallia kuvaavat pisteet voivat olla yhdistettynä toisiinsa suorilla jotka ovat käyriä. Tämä mallinnustapa voi siis mallintaa kaartuneita tai mutkikkaita pintoja vähemmällä määrällä pisteitä. Pisteiden välillä sijaitseva käyrä määrittyy sen pisteille annetuista erilaisista painoarvoista jotka vaikuttavat käyrän muotoon. Mitä suurempi määritelty painoarvo annetulla pisteellä on, sitä enemmän siinä kiinni oleva käyrä on vetähtynyt sen puoleen. (Ghosh, 2014).

2.2 Digitaalinen Sculptaus

Digitaalisella tai 3D – sculptauksella referoidaan tapaan mallintaa 3D objekteja, siihen tarkoitetulla ohjelmistolla. Tämä tekniikka mahdollistaa 3D objektin manipuloimisen tavoilla joita voisi hyödyntää esim. oikean saviobjektin muotoilemisessa (venyttää, nipistää, tasoittaa yms.). Tämä mahdollistaa huomattavasti vapaamman luomisprosessin kuin mitä ns. perinteiset 3D-mallinnus ohjelmat tarjoavat. (Stylianos, Vlachaki 2017).

Sculptaus ohjelmistojen vapaamuotoisesta muokkaustyylistä johtuen se mahdollistaa hyvin korkeatasoisten ja tiheitten meshien luomisen, joka taas johtaa hyvin yksityiskohtaiseen lopputulokseen. (Wikipedia-b, n.d) Tästä johtuen sculptaus ohjelmistot ovat yleistyneet varsinkin elokuva- ja peliteollisuudessa ja muissa aloissa joissa fotorealistinen lopputulos on päämääräinen tavoite. Sculptaus-ohjelmistojen käyttämä geometria 3D objektin esittämiseen vaihtelee ohjelmasta ohjelmaan. Suurin osa tällä hetkellä olevista ohjelmistoista markkinoilla hyödyntää ns. ”Mesh” pohjasta geometrian muokkausta, jossa muokattava objekti esitetään polygoneista koostuvasta mesh-verkostosta. Vaihtoehtoisesti, jotkut ohjelmistot hyödyntävät ”Voxel”-pohjaista geometriaa. Voxel-geometria perustuu objektin esittämiseen sen volyymin mukaan. Toisin kuin esim. polygon mesh-pohjaiset mallinnusjärjestelmät, voxelit eivät omista varsinaisia koordinaattitietoja vaan ne paikannetaan suhteessa toisiin voxeleihin. (Wikipedia-c, n.d).

Mesh-pohjaisten muokkausohjelmistojen vahvuus piilee niitten kyvyssä tukea useampia muokkaustarkkuuksia yhdessä objektissa. Jotkut muokattavan objektin osa-alueet voivat olla hyvin yksityiskohtaisia ja tiheitä polygoni määrältään, kun taas toiset osat objektista voivat olla hyvin yksinkertaisia ja sisältää hyvin isoja yksittäisiä polygoneja. Mesh mallinnuksen heikko puoli piilee sen rajoittuneen topologian kannalta. Mesh-pohjaisissa mallinnusjärjestelmissä. Tietty määrä muokattuja tai lisättyjä polygoneja mallissa saattaa rajoittaa muokausvapauttasi mallin myöhemmässä työstövaiheessa (liiallinen polygonien muutos / lisäys saattaa tehdä mallista rikkiäiset ja mesh kääntyy esim., sisäänpäin). (O’Rourke, 2012).

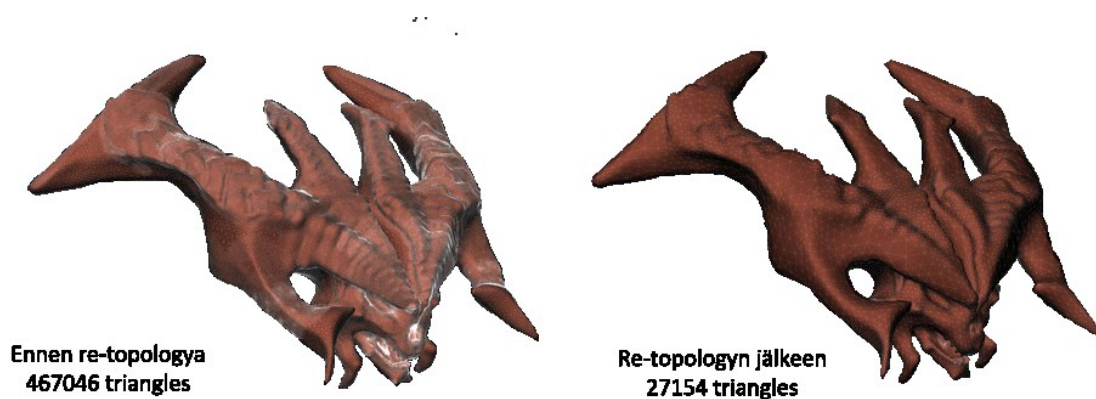
Vastaavasti voxel-pohjaisen mallinnusjärjestelmän hyöty piilee sen täydessä mallin muokausvapaudessa. Voxeleita hyödyntäessä malli ei voi ns. rikkoutua, kuten mesh-tyyppisen muokkauksen kanssa. Toisaalta voxeleita hyödyntäessä käyttäjän tulee olla tarkkana koska voxel-pohjaiset mallinnusjärjestelmät käsittelevät useita tarkkuuksia yhden mallin sisällä hyvin huonosti. Voxel mallia muokatessa tietyllä tarkkuudella, kaikki sitä pienemmällä tarkkuudella tehtyt aikaisemmat yksityiskohdat katoavat. (Ghosh, 2014).

Omasta kokemuksestani, vaikkakin sculptaus mahdollistaa hyvin korkealaatuisten ja upeitten mallien luomisen, nämä mallit ovat yleisesti liian vaativia tiheydeltään ja kooltaan että niitä voisi käyttää missään muussa tarkoituksessa. Kuitenkin oikeilla työkaluilla ja tekniikoilla niiden topologiaa voidaan muokata vähentämällä taktisesti polygonien määrää jotta mallista tulisi kevyempi käyttää mutta sen fotorealistisuus säilyy. Tämä on hyvin olennainen osa sculptausta ja siihen syvennyttään myöhemmässä osassa tätä opinnäytetyötä.

2.3 Re-topology

Re-topology:lla viitataan mallin muodon, tai topologian muuttamiseen. Re-topology on hyvin olennainen osa 3D-mallien muodostamisessa, varsinkin sculptausta käsitellessä. Kuten aikaisemmassa osiossa mainittiin, vaikka sculptaus mahdollistaa hyvin yksityiskohtaisten ja fotorealistisien mallien luonnin, niiden lopulliseen käyttöön ottoon niitä tulee hieman jälkikäsitellä. Re-topology on työprosessi jolla käsittelemättömät hyvin yksityiskohtaiset, ja laitteilta paljon vaativat 3D-mallit käsitellään niiden topologiaa yksinkertaistamalla, jotta polygonien määrä vähentyisi ja mallit eivät vaatisi niin paljon resursseja käytettävältä alustalta. Re-topology mahdollistaa siis mallien teknisen keventämisen, kuitenkin säilyttäen realismin tunteen mallin esteettisyydessä. (Martin, 2016). Re-topology voidaan suorittaa usealla eri tavalla ja eri työkaluilla, riippuen käytettävästä ohjelmistosta.

Kuten esimerkistä näkee, (Kuva 1) malli joka on käynyt re-topology työvaiheen, on säilyttänyt lähes täydellisesti muotonsa ja olennaisimmat yksityiskohtansa. Kuitenkin sen polygonien määrä ja tiheys on vähentynyt huomattavasti tehden siitä kevyemmän käsitellä myöhemmissä työvaiheissa. Polygonien määrä visualisoituu valkoisena mesh verkostona mallin päällä joka kuvaa mallin topologiaa.



Kuva 1. Esimerkki re-topologyn toimintaperiaatteesta

Riippuen käytettävästä alustasta, käytettävien polygonien määrä per hahmo tai objekti vaihtelee huomattavasti. Teknologian ja pelien alustojen nopeasta kehityksestä johtuen, monimutkaisempien ja polygonirunsaampien mallien käyttö on mahdollistunut. Re-topology käsittely on äärimmäisen olennaista erityisesti pelien kannalta. Koska pelit ja erityisesti pelimoottorit joutuvat renderöimään pelin sisäisiä objekteja ja hahmoja jatkuvasti pelin ollessa käynnissä, objektien optimoimisesta tulee hyvin tärkeää että pelin kulku olisi mahdollisimman sulava ja keskeytymätön. Elokvateollisuudessa, tämä ei ole yhtä olennaista sillä elokuvat ovat luonteensa mukaan pre-renderöityjä ja niiden ainoa rajoite hahmojen vaativuutta mielellä pitäen on aika, joka elokuvan renderöimiseen menee. Tämän saman erittelyn voimme nähdä myös videopeleissä, ja niiden väläanimaatioissa. On hyvin ymmärrettävää että pelien väläanimaatiot ovat usein paljon vaativamman näköisiä ja yksityiskohtaisempia kuin pelien varsinaiset grafiikat, koska

ne ovat nimenomaan pre-renderöityjä eikä niitä tehdessä tarvitse huolehtia yhtä paljon mallien re-topology käsittelystä. (Pluralsight, 2014).

2.4 Pelimoottori

Pelimoottori on pohjimmiltaan runko- tai kehys jolle peli on tarkoitus rakentaa. Ennen pelimoottorien kehitystä, pelien teko oli hyvin rajoittunutta. Monimutkaisempien ja vaativien ratkaisujen teko oli lähes mahdotonta senaikaisten laiterajoitusten myötä. Vaikka jotkin laiteratkaisut mahdollistivatkin hieman monimutkaisempien ja vaativampien pelien kehityksen, koska teknologian kehitys oli räjähtävän nopeata, varsin peliteollisuuden saralla, kehittäjät pääsivät hyödyntämään kokoajan uudempia ja hienompia resursseja peliensä kehittämiseen. Tämä johti puolestaan taas siihen että vanhempien pelien lähdekoodin uudelleenkäyttö oli mahdotonta kehittäjien siirtyessä alustalta toiselle teknologian kehityksen mukana. Pelimoottorien kehitys muutti koko alan katselmuksen tähän toimintatapaan ja mahdollisti aivan uudenlaisen toimintaprosessin, sekä pelienkehittäjille että muille peliartistille. (Wikipedia-d n.d) Pelimoottorien kehityksen myötä, pelientekijöitten ja peliartistien kädet vapautuivat varsinaisten pelien kehityksen rakentamisesta, niihin asioihin jotka tekisivät heidän peleistään uniikkeja. (Ward, 2008).

Pelimoottorin tarkoituksen on tarjota kehittäjille valmiit toiminnallisuudet jonka ympärille peli voidaan rakentaa. Pelimoottorien toimintaa voidaan verrata autojen toimintaan. Kaikki autot toimivat enemmän tai vähemmän samalla periaatteella. Autosta tekee uniikin sen omat itseään määrittelevät erilaiset osat kuten vaikka nahkapenkit, tai CD-soitin, kojelaudan toiminnot. Nämä osat tai toiminnot ovat autossa vastaavat, kuin mitä pelikehittäjät tekevät pelimoottorien kehukseen tehdäkseen peleistään uniikit. (Ward, 2008).

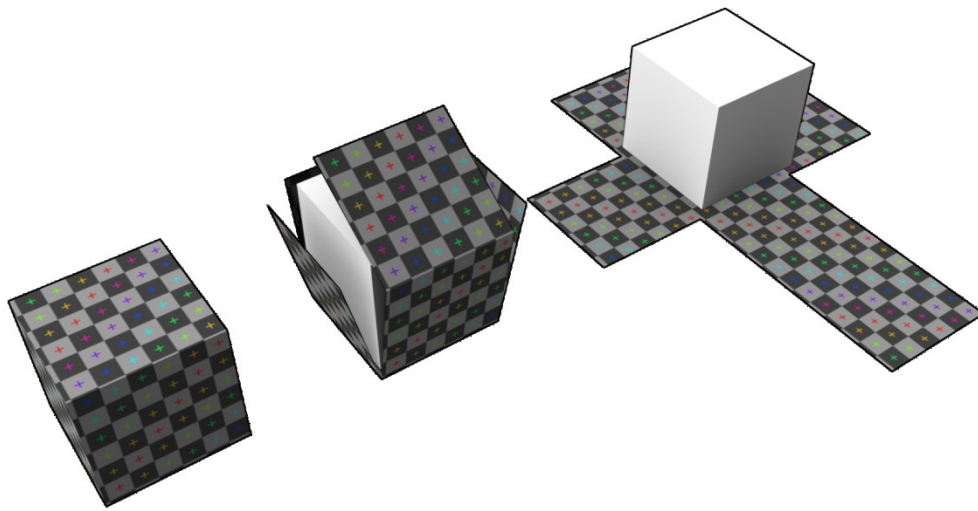
Pelimoottori pitää sisällään useita eri osia jotka muodostavat toimivan kokonaisuuden jolla peli voi myöhemmin pyöriä. Näitä osia ovat renderöintimoottori, joka on vastuussa 3D-grafiikkojen generoimisesta, fysiikkamoottori jota käytetään fysiikan lakien implementoimiseen peliympäristöön sekä mahdollinen tekoälymoottori joka mahdollistaa tekoälyn implementoimisen peliympäristöön. (Baker, 2016).

2.5 Teksturointi / UV-mapping

3D-mallien teksturointi perustuu työprosessiin nimeltä UV-kartoitus. UV-kartoituksella tarkoitetaan lyhyesti selitettynä kaksiulotteisen kuvan kartoittamista ja projektoimista kolmiulotteisen mallin päälle. ”UV-mapping” nimessä kirjaimilla U- ja V kuvataan klassisen koordinaatiston X ja Y akseleita. UV-kartoitus on äärimmäisen kätevä työkalu 3D-mallien luomisessa, ja mahdollistaa erilaisten pintaelementtien luomisen kaksiulotteiseen kuvaan ja niiden implementoimisen kolmiulotteisen mallin päälle. Kuva joka implementoidaan 3D-mallin päälle lopuksi, kutsutaan UV-tekstuuri kartaksi (UV-texture map). (Autodesk, 2014).

UV-kartoitus prosessi toimii yleisesti määrittämällä 3D-mallin pinnalta (usein polygonikohtaisesti) pisteitä tai kolmioita, jonka jälkeen 3D-mallin kartoitetusta pinnasta

luodaan kaksiulotteinen kuva (Kuva 2). Luotua karttaa voi käyttää tämän jälkeen pohjana halutun tekstuurin tai kuvan luomiselle. Valmis kuva / tekstuurikartta voidaan tämän jälkeen projektoida takaisin mallin päälle jonka jälkeen tuloksena on teksturoitu 3d-malli. Yleisesti mallin kartoittaminen tapahtuu automaattisesti ohjelmiston toimesta, vaikka se on mahdollista suorittaa myös manuaalisesti. Usein käytännöllisintä on hyödyntää ohjelmiston omaa kartoitusprosessia jonka jälkeen tekstuurikarttaa voi manuaalisesti muokata. (Wikipedia-e n.d).



Kuva 2. Esimerkki UV-mappingin toimintaperiaatteesta (Wikipedia n.d)

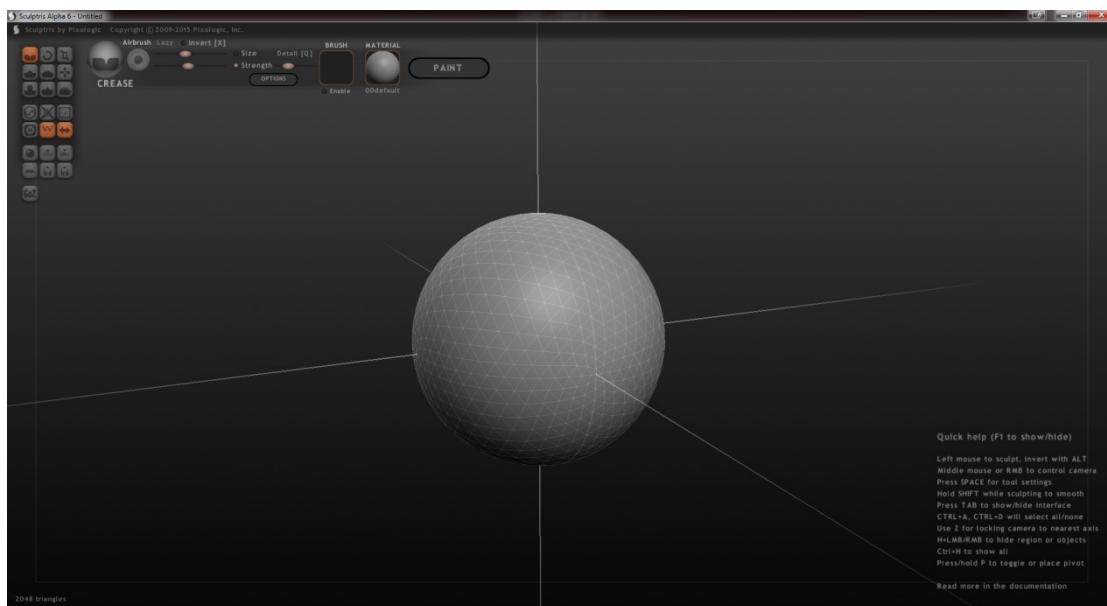
3 OHJELMISTOT

Tässä osiossa perehdytään hieman erityisesti tässä työssä käytettyihin ohjelmistoihin. Luoden ymmärrystä minkälaisessa ympäristössä malli on luotu, miten ohjelmistot ovat osallisena mallin luomisprosessissa ja millä tavoilla ne ovat vaikuttaneet lopputulokseen.

3.1 Mallinnusohjelmisto / Sculptris

Sculptris, on Tomas Petterssonin valmistama, nykyään Pixologicin omistama 3D-mallinnusohjelma joka on erikoistunut digitaaliseen sculptaukseen, ja on tämän opin näytetyön olennaisin työkalu. Sculptriksen ensimmäinen versio julkaistiin 2009, ja uusin julkaisu on tullut ulos 2011, mutta ohjelmisto on vielä kehityksen alla. (Wikipedia-a n.d). Sculptris on suunniteltu aloittelijoille ja on täysin ilmaiseksi ladattavissa Pixologicin omilta kotisivuilta.

Sculptris on verrattuna toisiin 3D-mallinnusohjelmiin hyvin yksinkertainen ja suoraviivainen käyttää, jonka takia se sopii erityisesti aloittelijoille. Ohjelmiston käyttöliittymä koostuu vain muutamasta osasta. Muokkainvalikko pitää sisällään kaikki vaihtoehdot millä pystyt kontrolloimaan tai hallinnoimaan 3D-malliasi (tässä tapauksessa, palloa) eri tavoilla (Kuva 3).



Kuva 3. Sculptris-ohjelman käyttöliittymä

Muokkainvalikko pitää sisällään päämääräiset mallin muokkaukseen tarvittavat työkalut. Koska sculptris ja muut digitaaliset sculptaus ohjelmistot koettavat simuloida oikeanlaista oikean elämän savimuokkausta tai kuvanveistoa, sen työkalut vastaavat paljon oikeita työprosesseja mitä voisi hyödyntää oikeassa elämässä. Sculptriksen mallinnus pohjautuu klassiseen polygonaaliseen mallinnustapaan, joka tarkoittaa että sculptriksen kaikki mallinnustyökalut perustuvat joko mallin polygonimeshin muuttamiseen

joko lisäämällä tai vähentämällä mallin massaa, tai muokkaamalla eri tavoilla jo valmiiksi olemassa olevaa mallin massaa eri tavoilla.

Sculptris pitää myös sisällään alkukantaisen paint toiminnon, joka mahdollistaa yksinkertaisten tekstuuri- ja bump karttojen luonnin jota voi myöhemmin hyödyntää luotujen mallien teksturoinnissa ja jälkikäsitteilyssä.

3.2 Pelimoottori / Unity

Unity on Unity Technologiesin kehittämä usean alustan pelimoottori, jota käytetään päämäärin pelikehitykseen mutta myös esimerkiksi simulaatioiden valmistamiseen. Unity (versio 1.0) julkaistiin 8.6.2005 mutta viimeisin versio pelimoottorista on julkaistu 17.12.2017. (Brodin 2013). Unitylla on mahdollista luoda sekä 2D- että 3D-pelejä tietokoneelle, konsoleille ja eri mobiilialustoille. Unity on aikoinaan tukenut useita eri koodauskieliä mutta kielet Boo sekä JavaScript poistettiin käytöstä moottorin sisällä 2017 aikana, jonka jälkeen moottorin päämääräiseksi koodauskieleksi jäi C#. (Fine, 2017). Unity on saatavilla ilmaiseksi heidän omilta kotisivuiltaan, mutta siitä on saatavilla myös erikseen, Plus ja Pro versiot, jotka ovat pääsijaisesti tarkoitettu ammattilaiskäyttöön.

Unity on hyvin laaja työkalu pelien kehitykseen, koska se pitää sisällään useita eri toiminnallisuuksia jolla pelejä voi luoda. Unityn avulla, pelikehittäjä voi luoda lähes kaiken tarvittavan peliin, peliympäristöstä ja sen ominaisuuksista animaatioihin, pelin audiomaailmaan ja jopa verkkotoiminnallisuudet mahdollista internet pelaamista varten. Unity pitää sisällään myös laajat muokkausmahdollisuudet muita objekteja varten joita unityn sisällä ei välttämättä pysty luotettavasti luomaan, kuten hahmojen tekstuurit ja niiden ominaisuudet. (Wikipedia-f, n.d).

Unityn materiaalin muokausmahdollisuudet perustuvat muokattavan materiaalin tekstuurien ja valaistusominaisuuksien säätämiseen. Materiaalit unityn sisällä määrittävät kuinka objektien pinnat renderöidään, referoimalla käytettäviin tekstuureihin, väreihin ja moniin muihin asioihin. Materiaalit ja niiden renderöinti riippuu myös hyvin paljon käytettävistä shadereista. Shaderit ovat toiminnallisuuksia jotka pitävät sisällään matemaattiset laskelmat ja algoritmit joilla lasketaan renderöitävien pixelien ominaisuudet, asetettujen parametrien perusteella (valaistus, materiaalit). Unity tarjoaa käyttäjilleen kattavat valikoimat shadereita eri tarkoituksiin: FX shader valaistukselle ja lasiefekteille, Nature maaston ja puiden renderöimistä varten, Particle erinäisiä hiukasefektejä varten ja monia muita. Materiaali- ja shadereitten kontrolloimisen lisäksi unityn sisäisiä objekteja voi muokata hallinnoimalla käytettäviä tekstuureja. Tekstuureja voi lisätä kuvamuodossa objektien päälle, jonka jälkeen hyödyntämällä erilaisia materiaaleja ja shadereita, objektien muokausmahdollisuudet ovat lähes rajattomat. (Unity-a, 2017).

Unityn pitää sisällään myös omat toiminnallisuudet erinäisten animaatioiden luomiseen. Unity tarjoaa käyttäjilleen helpon ja käyttäjäystävällisen hahmo- ja objektianimaatioiden luomisprosessin sekä mahdollisuudet animaatioiden helppoon hallinnoimiseen ja yksityiskohtaiseen muokkaamiseen hyödyntäen erilaisia työkaluja. Unity tukee esimerkiksi useitten eritasoisten animaatioiden tapahtumisen samassa objektis-

sa, (eri animaatio, eri ruumiinosissa eri logiikalla), mahdollistaa eri animaatioiden interaktion muokkaamisen toistensa kanssa, sekä unityn ulkopuolisten animaatioiden tuonnin ja muokkaamisen sisään ohjelmaan. Unity tarjoaa mahdollisuuden hahmojen animaattoreille työskennellä irrallaan varsinaisesta pelikehitysprosessista. (Unity-b, 2017).

Unityn sisällä on mahdollista myös hallinnoida pelien tietoverkko puolta. Unityn oma API tekee käyttäjille mahdolliseksi luoda internetin kautta toimivia moninpelejä. Unityn oma "High level API" (HLAPI) auttaa käyttäjiä rakentamaan tarvittavat ominaisuudet peleilleen huolehtimatta alemman tason networking ominaisuuksien implementoinnista. Unityn HLAPI:n avulla käyttäjät voivat simuloida varsinaista internetin yli käytävää moninpeli kokemusta ja esimerkiksi lähettää networking komentoja, tapahtumia sekä viestejä rakentaessaan monin pelejensä. (Unity-c, 2017).

4 KÄYTÄNNÖN PROJEKTI

Tässä osiossa työtä käydään läpi käytännön prosessia 3D-mallin luomisesta, sen teksturoimisesta ja tuomisesta pelimoottoriin. Tavoitteena on aste asteelta tutkia 3D-hahmon mallinnusprosessia, miten sculptaus-ohjelmiston sisällä itse mallinnustyö tapahtuu ja minkälaisilla työkaluilla. Tämän jälkeen mallille on tarkoitus suorittaa re-topology jälkikäsittely jonka tehtävänä on muokata mallin polygonirakennetta, säilyttäen tärkeimmät yksityiskohdat, mutta karsien liiallista polygonitiheyttä tehden mallista kevyemmän käsiteltävän. Re-topology käsittelyn jälkeen, malli teksturoidaan ja valmistellaan pelimoottoria varten, jonka jälkeen hoidetaan viimeiset jälkikäsittelyt pelimoottorin sisällä. Pelimoottorin sisällä mallia hienosäädetään säätämällä mallin käyttäytyminen valaistuksessa, sekä mahdolliset mallin toiminnallisuudet esim. karvoitus tuoden projektin päätökseen.

Projektin asteittaisen tarkastelun lisäksi, tarkoituksena on kiinnittää huomiota työn mahdollisiin ongelmakohtiin; mitä toimintatapoja kannattaa välttää, mitkä asiaseikat voivat johtaa projektin kokonaisuuden epäonnistumiseen ja kuinka näitä asioita voidaan mahdollisesti ratkaista. Opinnäytetyön käsittely ympäristö on päämäärin tässä työssä esitelty ohjelmistot.

4.1 Mallin sculptaus

Ensimmäinen työvaihe projektissa on mallin sculptaaminen. Sculptaaminen koostuu itsessään useista eri työvaiheista, joita voi tarkastella projektin työkulua kuvaavassa kaaviossa (Kuva 4).



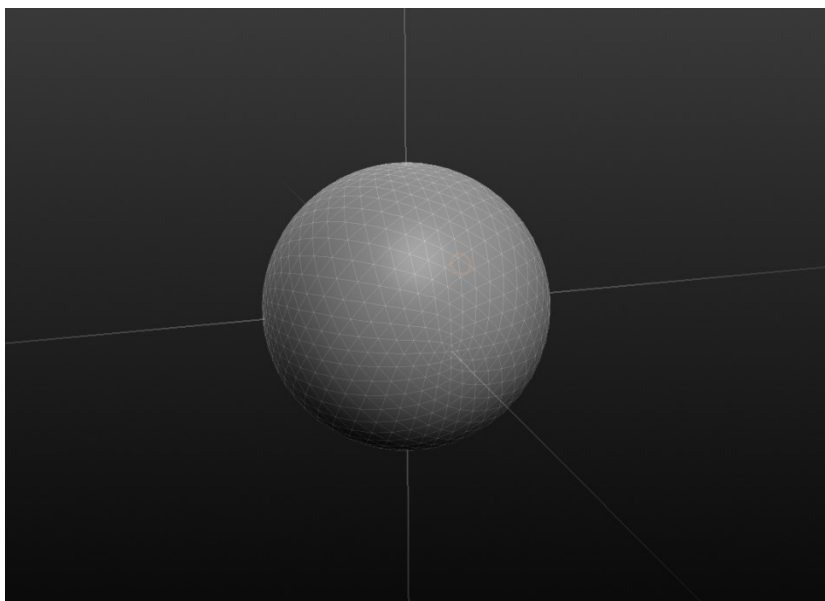
Kuva 4. Projektin työnkulku, sculptaus.

Tässä projektissa on tarkoitus luoda pelihahmo pelimoottorikäyttöä varten, mutta erityisesti PC, tai konsolialustoja miettiessä. Malli tulee olemaan päämääräisesti ihmismäinen mutta pienillä eläinmäisillä piirteillä, tarkoitettu fantasiamaiseen peliympäristöön. Pelimoottorin rajoituksia mielellä pitäen, malli tulee sculptata asentoon jossa sen sisälle on helppo asentaa luuranko myöhempiä animointia varten. Mallin animointi ei ole osa tätä projektia, mutta mallin luominen myöhempiä työprosesseja mielellä pitäen on hyvin tärkeää 3D-malleja luodessa.

Käytetyn sculpting-ohjelmiston Sculptriksen lähtöpiste jokaiselle mallinnukselle on yksinkertainen pallo (Kuva 4.). Ohjelmisto tukee myös useita muokattavia objekteja samaa aikaa, mutta koska suunniteltu malli koostuu vain yhdestä yhtenäisestä objektista, lisäobjektit eivät ole tarpeellisia.

4.1.1 Muodon hahmottelu

Mallista halutaan muokata aluksi jonkinlaisen ihmismäisen näköisen mallin josta voi työstää eteenpäin yksityiskohtia (Kuva 5). Koska työstössä on vasta mallin yleinen olo-muoto, on olennaista muistaa hyödyntää sculptriksen symmetry toimintoa, joka jakoi mallin kahtia ja mahdollistaa mallin kummankin puolen identtisen mallintamisen yhtä aikaa.



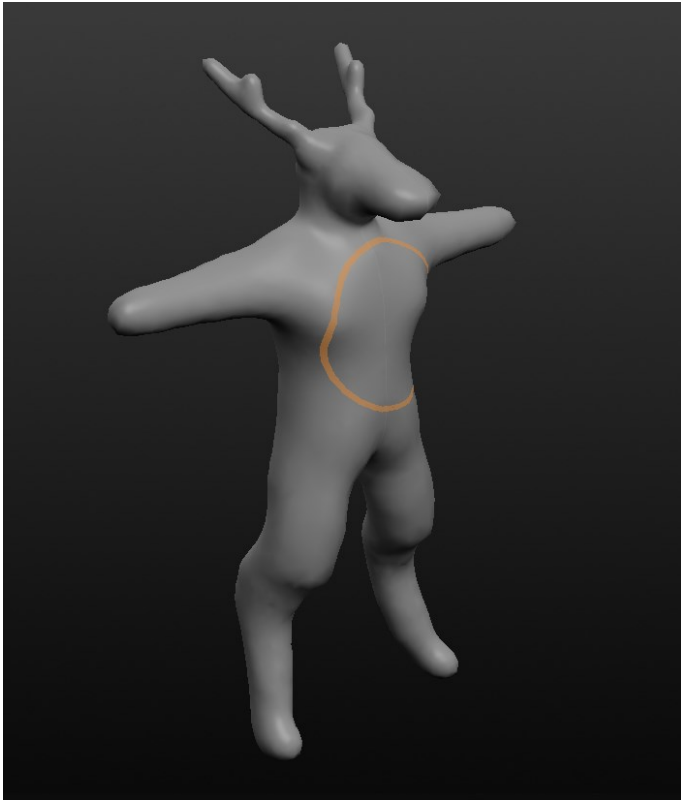
Kuva 5. Lähtöpiste mallinnukselle

Mallin yleisen muodon mallintamiseen käytettiin enimmäkseen inflate toimintoa. Saman lopputuloksen olisi voinut saada muillakin toiminnoilla (esim. draw), kunhan ulokkeiden luominen tapahtuu ns. "lisäämällä" materiaalia alkuperäiseen palloon, eikä venyttämällä pallon alkuperäistä materiaalia. Sculptriksen mesh-pohjaisen geometrian takia, polygonien liiallinen muokkaus voi tuhota mahdollisuudet myöhempiin muokkauksiin jonka takia on hyvä luoda lisää muokattavia polygoneja eikä venyttää jo olemassa olevia.



Kuva 6. Päämääräiset ulokkeet (kädet, jalat, pää) hahmoteltuna.

Hahmon mallintaminen jatkuu hyvin samoilla linjoilla tästä eteenpäinkin, lisäämällä materiaalia kunnes mallin haluttu ulkomuoto vastaa odotuksia siihen pisteeseen että siihen voi alkaa muokkaamaan yksityiskohtia (Kuva 7). Tässä vaiheessa mallinnusprosessia oli tärkeitä tehdä jatkuvasti pieniä lisäyksiä ja pieniä korjauksia koko vaiheen aikana, aste asteelta. Esimerkkinä, on huomattavasti helpompaa rakentaa hahmolle kädet ja jalat suhteessa vähitellen oikeaan suuntaan jatkuvasti hioen ja korjaillen, kuin luoda epämääräinen massa lähes oikeaan suuntaan jota pitää myöhemmin korjata ja muokata usealla eri tavalla ja hyvin radikaalisti. Tämä on hyvin tärkeä seikka, varsinkin sculptriksen kanssa, koska ohjelmisto sallii vain tietyn määrän muokkausten kumoamista ctrl + alt + z toiminnolla jonka jälkeen ainoa keino saada malli aikaisempaan olomuotoon on uudelleen ladata tallennettu tiedosto. Jos jostain syystä et ole muistanut tallentaa tiedostoasi eri aikoina, on hyvin mahdollista että suurin osa tehdystä mallinnustyöstä pitää tehdä uudestaan.



Kuva 7. Lopullisen ulkomuodon hahmottaminen.

Mallin karun ulkomuodon mallintaminen tapahtui enimmäkseen lisäämällä draw tai inflate työkalulla lisää materiaa johonkin paikkaan, jonka jälkeen käytin smooth työkalua mallin hiomiseksi oikeanlaiseen muotoon, jonka jälkeen grab työkalulla sai vielä hienosäädettyä mallinnettua pintaa. Kun olennaisimmat hahmon ulokkeet ovat luotu, aloin hiomaan niitä tarkemmiksi. Mittasuhteet vaihdettiin kohdilleen liikuttamalla ja venyttämällä raajat ja pään grab työkalulla, joka oli myös kätevä työkalu pintojen muotojen muokkaamisessa. Suurimmat kropan muotojen ja kallon muotojen luominen onnistui juuri nimenomaan grab työkalua hyödyntäen. Kun mallin yleinen ulkomuoto oli hoidettu valmiiksi, oli aika aloittaa yksityiskohtien luominen. (Kuva 8)



Kuva 8. Mallin päämääräinen ulkomuoto valmiina.

4.1.2 Mallin yksityiskohdat

Hahmolle tuli luoda ihmismäinen lihasrakenne ylävartaloon, sekä eläinmäiset jalat. Pääksi hahmolle tuli laittaa hirvieläimen kallo sarvineen. Lihasten luominen onnistui parhaiten draw toiminnolla joka mahdollisti lihasten piirtämisen suoraan kehoon. Hyödyntämällä draw toiminnon clay vaihtoehtoa lihasten työstämisestä tuli vielä helpompaa. Ilman clay vaihtoehtoa, draw toiminto jatkaa massan lisäämistä kursorin kohdalle jatkuvasti niin kauan kun se on aktiivisena. Clayn ollessa aktivoituna, draw toiminto lisää massaa samalla tavalla kuin aiemminkin mutta ei reagoi jos kursori liikkuu jo alueen yli mihin on lisätty massaa samalla hiiren painalluksella. Lihaksia mallintaessa hyödynnettiin myös hyvin aktiivisesti crease toimintoa korostamaan ja erittelemään lihaksia toisistaan. Crease toiminnolla luotiin lihaksille ääriviivat jonka jälkeen käytettiin smooth toimintoa pehmentämään hieman jälkeä, koska crease toiminnolla luotu taitos on yleisesti heikoimmillaankin hyvin vahva. Näin lihaksista saatiin tehtyä hieman luonnollisemman näköiset.

Sarvien luonti alkoi luomalla alkukantaiset ulokkeet sarven erkaantuville piikeille. Kun hyvin pienet tyngät sarvien ulokkeille oli luotu draw toiminnolla, niitä venytettiin grab toiminnolla hienovaraisesti haluttuun suuntaan ja muotoon. Venyttämällä, sarvet saivat pallomaisten tynkien muodon sijasta hieman terävämmän ulkomuodon. Sarvien terävyyttä hiottiin vielä hieman pinch toiminnolla joka nipisti kärjet valmiiksi (Kuva 9).



Kuva 9. Yksityiskohtien luomista (lihakset, sarvet, sormet, kallon muoto).

Tulevia mallin tekstuureja mielessä pitäen, myös pienimuotoiset rajat eri tekstuurien välille luotiin. Mallin rinta- ja vatsalihakset, kainalot, hauikset, kämmenet sekä kaula tulisivat olemaan eri tekstuuria kuin loppukeho joten erityisesti rajapintoja korotettiin näiden alueiden välillä. Tämän lisäksi myös kallo sekä jalkojen kaviot tulisivat olemaan eri tekstuuria kuin loppukeho joten niitten välille luotiin myös hienovaraiset rajat. Rajojen luonti onnistui samalla tyyllillä kuin lihasten rajausta, eli crease toiminnolla ja smooth toiminnolla hienontaen jälkeä. Sen lisäksi kehon pinnan korkeutta säädettiin verrattuna rintalihaksiin, korostaen hieman eri rajapintojen eroa hahmossa.

Yksityiskohtia luodessa oli hyvä pitää mielessä että muokattavat polygonit eivät pysty mahdottomuuksiin, varsinkin sormia ja sarvia mallintaessa. Pieniä ulokkeita mallintaessa ja varsinkin niiden käsittelyssä oli olennaista tarkastella välillä wireframe näkymällä että muokattavia polygoneja oli tarpeeksi. Tämä on tärkeää, koska myöhemmässä työvaiheessa kun mallin topologiaa jälki käsitellään, on tärkeää että mikään hahmon osa voi vääristyä käyttökelvottomaksi. Varsinkaan kun sculptriksen undo toiminto on hyvin rajoittunut, eikä sillä voi kumota tehtyjä muutoksia hirveän pitkälle takaisinpäin.

Sen jälkeen kun malli oli rakenteellisesti valmis, ja suurimmat yksityiskohdat olivat valmiina, symmetryn voi kytkeä mallista pois päältä. Tämän jälkeen viimeiset uniikit yksityiskohdat luotiin malliin (Kuva 10), kuten sarvien erottelemisen erilaisiksi ja käsien ja jalkojen muokkaus sekä kallon yksityiskohtien hionta. Tarkoituksena oli luoda mallin erinäisiä osia hieman erilaisiksi, koska mikään hahmo ei ole luonnostaan ikinä täysin symmetrinen.



Kuva 10. Valmis sculptattu 3D-malli.

4.2 Re-topology

Kun 3D-malli on viimeistelty, seuraavana työvaiheena on re-topology. Re-topology käsittelyn työvaiheet voidaan nähdä kuvasta 11 joka visualisoi projektin työkulua. Re-topology käsittely hoidetaan kuvan mukaisesti: Ensin suunnittelemalla ja hahmottamalla haluttu polygonitavoite ja miten tavoitteeseen on helpointa ja käytännöllisintä päästä. Tämän jälkeen käsittely toteutetaan mallille eri asteittain.



Kuva 11. Projektin työkulku, siirtyminen re-topologyyn.

4.2.1 Suunnittelu ja tavoitteet

Kuten projektissa on aikaisemmin esitetty, re-topologyn tarkoituksena on vähentää polygonien määrä mahdollisimman pieneksi, kuitenkin säilyttäen tärkeimmät 3D-mallin yksityiskohdat. Sculptriksen omat työkalut polygonimäärän vähentämiseksi toimivat päämääräisesti manuaalisesti, mutta pitävät sisällään myös vaihtoehdon vähentää tai kasvattaa polygonitiheyttä universaalisti koko mallin alueelta, mutta koska mallinnettu hahmo pitää sisällään useita eri polygonitiheyksiä eri alueilla, emme halua hyödyntää kattavia yhtenäisiä työkaluja.

Mallin käsittelyn jälkeistä polygonitiheyttä mielessä pitäen, tämän työn kannalta tarkoituksena on päätyä noin parinkymmenen tuhannen polygonin alueelle. Haluamme alen-
taa mallin polygonitiheyttä niin paljon kuin pystymme, mutta hyvänä numeerisena arvona, haluamme asettua lopputuloksen kanssa vastaavanlaiseen tulokseen, kuin mitä seitsemännen sukupolven konsolit (Playstation 3, XBOX 360) toivat esille peleis-
sään. Useat vastaavien konsoleitten pelien grafiikoista pyörivät vastaavanlaisella alu-
eella polygoneja miettien, joten tarkoituksena on saavuttaa vastaavanlainen lopputu-
los tätä projektia ajatellen; suhteellisen fotorealistinen. Kuvasta 12 näemme projektin
viimeistellyn mallin wireframe näkymässä joka visualisoi 3D-mallin polygoneja valkoi-
sella rautalankanäkymällä, sekä ohjelmiston kirjaaman polygonien määrän kolmioina
(257906).

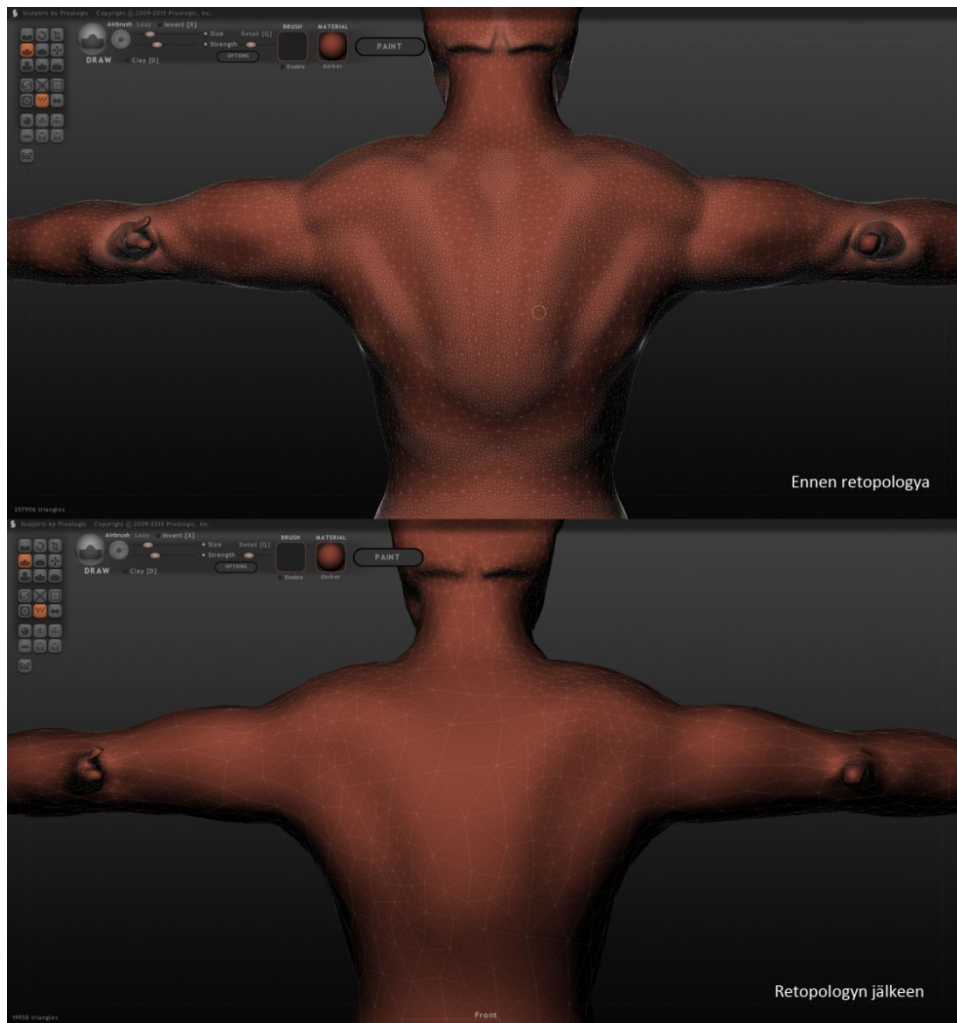


Kuva 12. Valmis 3D-malli wireframe näkymässä

Mallin re-topology aloitettiin työstämällä ensin laajat alueet jotka eivät pidä sisällään tarkkoja yksityiskohtia kuten, selän ja jalat, josta siirryttiin vähitellen yksityiskohtaisempiin osiin jotka vaativat enemmän hienovaraisempaa käsittelyä.

4.2.2 Käytännön käsittely

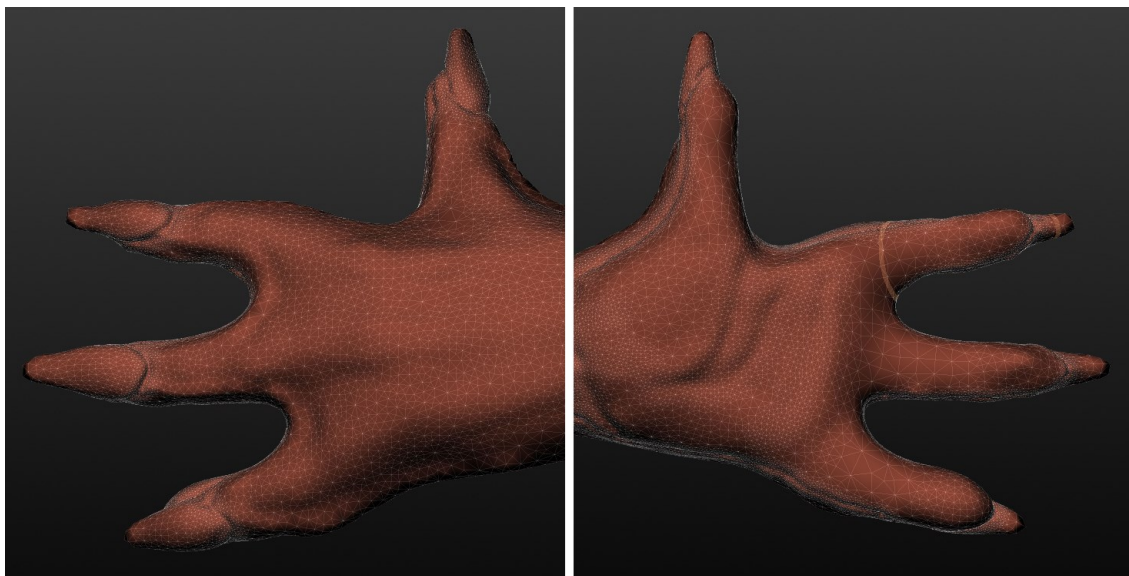
Selän, jalkojen ja kylkien käsittely oli hyvin yksinkertaista, koska niissä ainoat olennaiset osat olivat päämääräiset lihasten muodot ja kehon sopusuhtaisuus. Kunhan re-topologaus ei karsinut niitä liikaa, polygoneja pystyi vähentämään suhteellisen rankasti, koska halutut muodot eivät vaatineet suuria määriä polygoneja kuvaamaan niitä halutusti. Kuvasta 13 näkyy kuinka re-topology käsittelyn jälkeen polygonien tiheys mallissa vähenee huomattavasti mutta halutut selkälihasten muodot säilyvät ennallaan.



Kuva 13. Vertailu mallin selästä ennen re-topologia ja sen jälkeen.

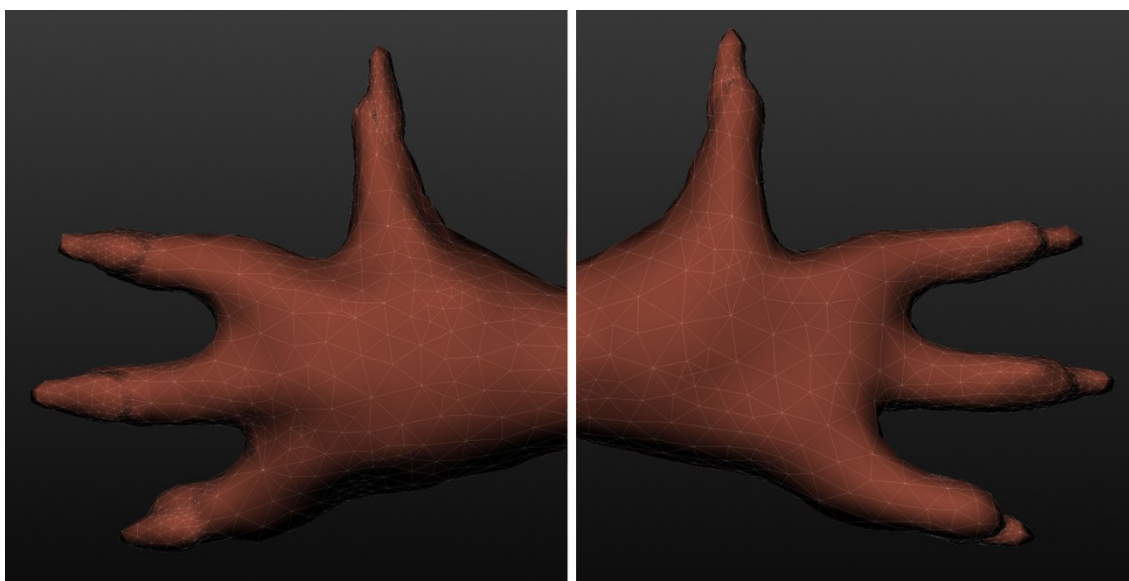
Hyvänä asiana muistaa mallin re-topology käsittelyssä on mittakaava jolla hahmoa tullaan tarkastelemaan sen varsinaisessa käytössä. Tämä kyseinen hahmo tässä projektissa on tarkoitettu pelin sisäiseen käyttöön jolloin hahmoa ei varsinaisessa käytössään todennäköisesti tulla tarkastelemaan äärimmäisen läheltä. Pitäen tämän mielessä, yksityiskohtia voidaan karsia erityisesti sellaisista paikoista jotka ovat joko niin pieniä, tai sellaisessa paikassa jossa pelin sisällä kyseisiä paikkoja ei tarkastella niin intensiivisesti. Tällaisia paikkoja tässä kyseisessä hahmossa ovat esimerkiksi sormet, sekä kynnet.

Kuvasta 14 näkyy wireframe näkymällä katsottuna kuinka tiheä mallin mesh on varsinkin käsien ja kynsien alueella. Nämä alueet ovat toki yksityiskohtaisia, koska ne koostuvat useista tarkoista kokonaisuuksista, mutta on selvää että käsien polygonitiheyttä voi karsia hieman rankemmalla otteella.



Kuva 14. Hahmon kädet ennen re-topology käsittelyä.

Erityisesti kynsien sekä sormien liitoskohdassa, mallin mesh on kahden pinnan jyrkän kulman ja luodun uurteen takia äärimmäisen tiheä, joten käsittelin näitä kohtia siihen pisteeseen että enempien muutosten tekeminen vaikuttaisi negatiivisella tavalla kynsien ja sormien muotoon. Tällä tavalla polygonitiheyttä sai suuresti vähennettyä mutta olennaisimmat yksityiskohdat ja mallin muodot säilyivät (Kuva 15).

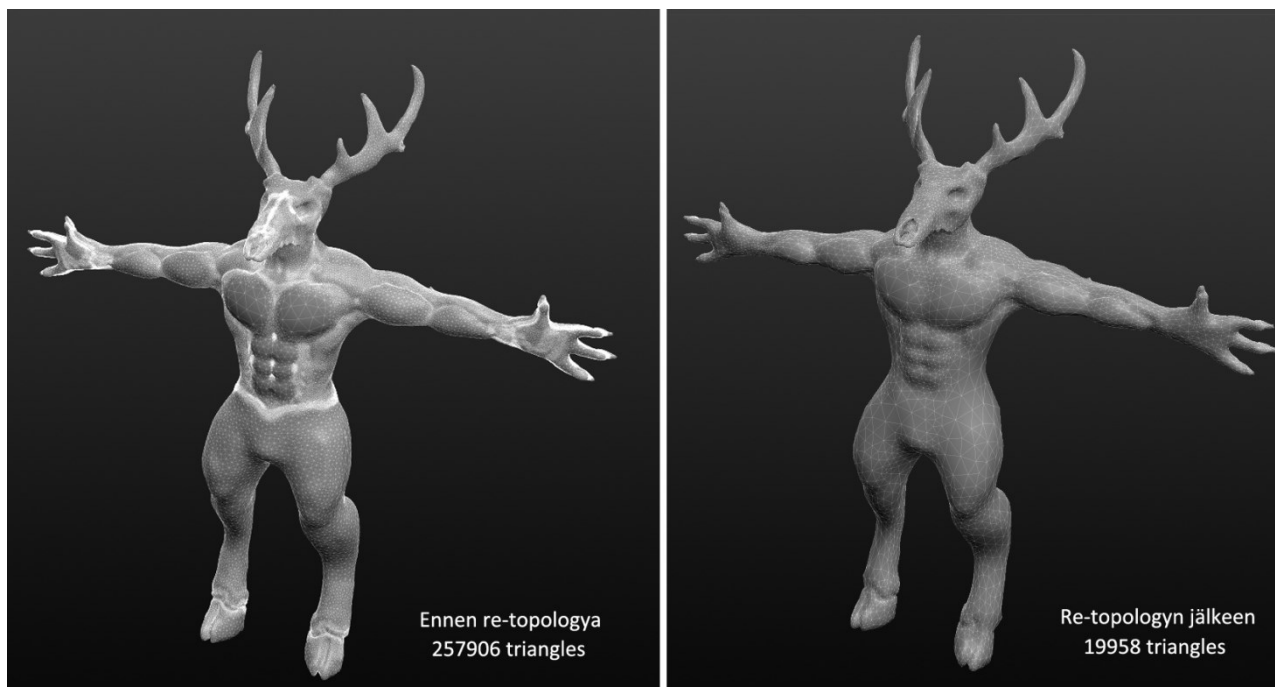


Kuva 15. Hahmon kädet re-topology käsittelyn jälkeen.

Ehkä olennaisin osa mallin re-topologia oli hahmon pääkallon käsitteleminen. Pääkallo oli pelkältä ulkomuodoltaan monimutkaisin yksittäinen alue, ja sen yksityiskohdat olivat hyvin tärkeitä säilyttää. Prosessi muuten oli sama, käyttäen reduce brushia polygonitiheyttä vähennettiin pisteeseen jossa muoto ja halutut yksityiskohdat säilyy mutta polygonitiheys vähenee huomattavasti. Olennaisimpia osia kallossa re-topologyn suhteen olivat erityisesti sarvet ja silmät.

Sarvia muokatessa oli syytä olla varovainen, sillä jos niiden teräviä kärkiä käsittelee huolimattomasti, ne voivat muuttua hyvin luonnottoman näköisiksi polygonitiheyden vähentyessä. Pääkallon silmät olivat erityisesti paikka josta oli olennaista vähentää polygonitiheyttä. Silmät tullaan teksturoinnin yhteydessä maalattua lähes mustaksi, ja koska ne ovat syvennykset kallossa, niiden polygonitiheyttä voi vähentää hyvin huomattavasti.

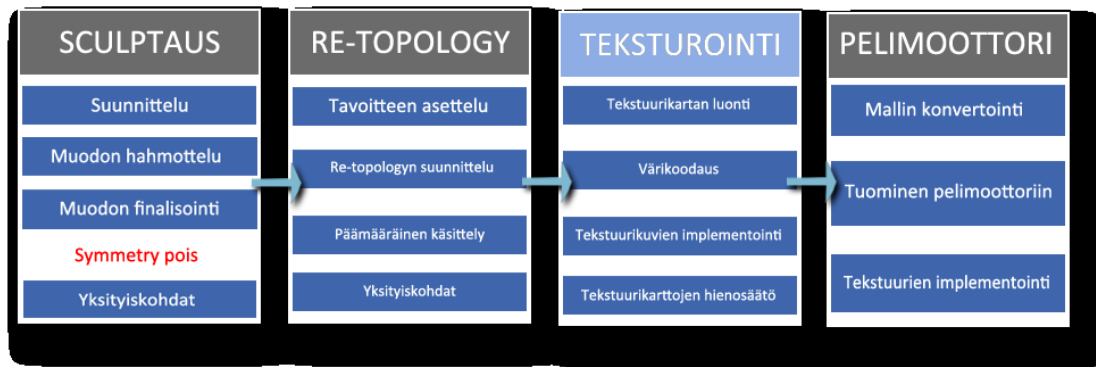
Kun malli on käynyt läpi re-topology käsittelyn, polygonimäärä on vähentynyt mallissa 257906 kolmiosta 19958 kolmioon, eli yli kymmenkertaisesti. Kuvasta 16 voimme nähdä kuinka mallin polygonitiheys on vähentynyt. Malleista näkyy karkeasti, mitä valkoisempi sitä suurempi polygonitiheys sillä alueella on, ja vertailusta näkyy selvästi kuinka paljon se on vähentynyt yleisesti sekä erityisesti tietyillä alueilla. Huomaamme myös että vaikka polygonitiheys on vähentynyt yli kymmenkertaisesti, malli on silti lähes samanlainen ulkomuodoltaan. Näin projektin noin parinkymmenen tuhannen polygonin tavoite on saavutettu.



Kuva 16. Vertailu mallista ennen re-topologia ja sen jälkeen.

4.3 Teksturointi

Kun mallin re-topology käsittely on valmis, seuraavana siirrytään mallin teksturoimiseen. Teksturoiminen onnistuu osittain sculptriksen omien toimintojen kautta, mutta tekstuurien muokkaaminen ja hienosäätäminen hoidetaan Adobe Photoshopissa. Kuvasta 17 näkyy teksturoinnin työvaiheet: Tekstuurikartan luonti, värikoodaus, tekstuurikuvien implementointi ja lopuksi niiden hienosäätö. Teksturoinnin työprosessi hoidetaan tässä järjestyksessä jonka jälkeen siirrytään työn pelimoottorivaiheeseen.



Kuva 17. Projektin työnkulku, siirtyminen teksturointiin.

4.3.1 Tekstuurikartan luonti ja maalaus

Kun re-topology käsittely on päätetty, siirytään sculptriksen sisällä paint tilaan. Tässä vaiheessa työtä on oltava varma että mallia ei halua sculptata enää, koska sen jälkeen kuin paint vaiheeseen on siirrytty, mallia ei yleisesti voi muokata hylkäämättä jo luotua paint dataa. Tässä vaiheessa työprosessia on siis hyvä tallentaa valmiiksi käsitelty malli ennen siirtymistä paint tilaan.

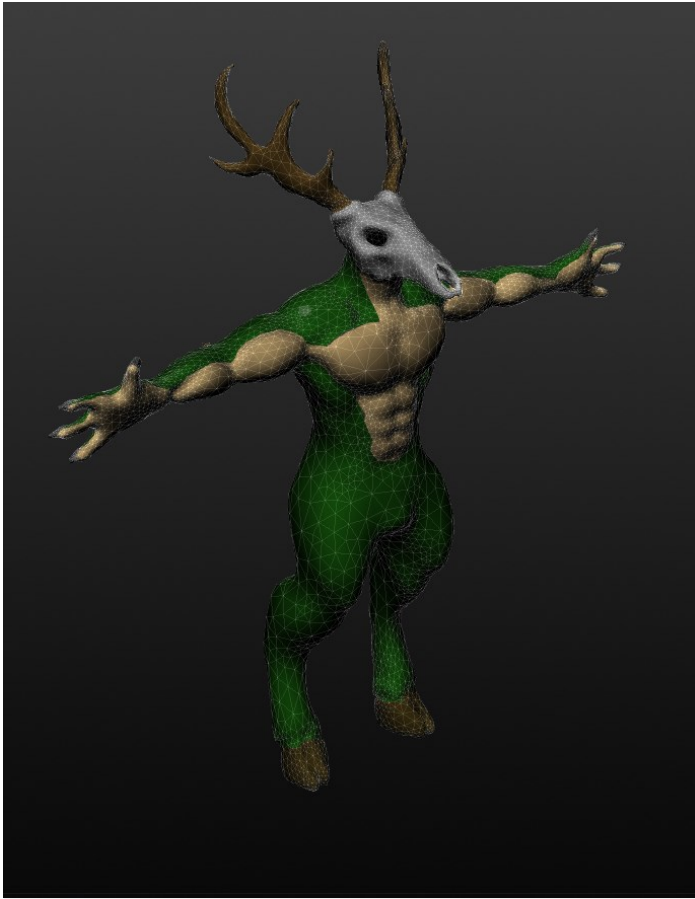
Sculptriksen omat ns. teksturointi mahdollisuudet ovat toistaiseksi hyvin rajalliset koska sen kehitys ei ole vielä lopussa, mutta se pitää sisällään tarpeeksi ominaisuuksia joilla pääsee hyvin alkuun mallin teksturointiprosessissa. Siirtyessä paint tilaan sculptriksessa, ohjelmisto luo 3D-mallista valmiin texture mapin jota tulemme käyttämään tekstuuriemme pohjana tässä työssä. Ohjelmiston luoma texture map on 3D-mallimme polygoniverkoston kaksikulotteinen versio. (Kuva 18).



Kuva 18. Sculptriksen luoma raaka texture-map.

Koska mallinnettu hahmomme on hyvin yksityiskohtainen, samasta syystä sen purettu kaksiulotteinen kartoitus on myös hyvin yksityiskohtainen. Seuraavana projektina on siis merkata hahmomme tavalla jolla ymmärrämme kartoituksen jaon myöhempää tekstuurien käyttöön ottamista varten. Tämä työprosessi hoidetaan sculptriksen paint-toiminnon kautta. Luomme siis jaottelun mallin eri osien välille ja väri koodaamme sen tavalla jolla ymmärrämme että mikä osa texture mapissa vastaa mitäkin kohtaa mallisamme. Haluamme jakaa värien puolesta mallin 6 eri osaan: Kroppaan, päähän, sarviin, kavioiden ja kynsiin, silmiin sekä rinta- ja käsien lihaksiin. Jokaiselle näistä osa-alueista tulee myöhemmin eri tekstuurityyppi, joten tässä vaiheessa projektia väritämme ne tunnistettavilla väreillä. Valitaan väreiksi: vihreä, (kroppa) beige, (lihakset) vaalean ruskea, (sarvet) tumman ruskea, (kaviot ja kynnet) musta (silmit) ja valkoinen (pää).

Maalaustyö sculptriksen sisällä on hyvin yksinkertaista. Ohjelmisto antaa mahdollisuudet eri värien valitsemiseen, eri voimakkuuksiin ja kokoihin ja materiaalin valitsemismahdollisuus on säilynyt ennallaan, jotta mallia voi tarkastella eri materiaalityypeillä esim. kiiltävänä tai mattana. Maalasin 3D-mallin näillä työkaluilla auttavalla tavalla antamaan suuntaa loppu projektille (Kuva 19).



Kuva 19. Sculptriksessa maalattu 3D-malli.

Kun malli on maalattu haluttuun pisteeseen asti, voimme tallentaa ohjelmistosta uudet texture mapit, jotka antavat jo paljon enemmän viittausta mallin koostumuksesta ja rakenteesta maalattujen värien toimesta (Kuva 20). Kuvasta 20 näkee jo selvästi miten mallin topologia on jaettu kaksiulotteiselle kartoitukselle. Tämän kuvan kanssa voimme siirtyä eteenpäin tekstuurien luomiseen.

Tekstuurien teko onnistuu luomalla kyseisen tekstuurikartan päälle haluamamme tekstuurit. Kun tekstuurit luodaan värikoodattujen osien päälle samaan tiedostoon ja tiedosto ladataan takaisin mallin päälle, tekstuurit tulevat mallin päälle kartan mukaisesti. Näin mallin päälle on ladattu auttava tekstuurikartta.



Kuva 20. Värikoodattu texture-map.

Maalatus tekstuurikartan perusteella kartan eri osat jaetaan mallista omille tasoilleen photoshopissa. Kun jokainen osa on erillisenä tasona kuvatiedostossa, niitä voi vapaasti käsitellä toisistaan riippumatta. Jakamisen jälkeen tekstuurit voidaan asettaa jokaisen erillisen värikoodatun alueen päälle. Mallia varten valittiin neljät eri tekstuurit jota hyödynnettiin mallissa.

4.3.2 Tekstuurikuvien luonti ja implementointi

Tekstuureja valitessa, on olennaista valita tai valmistaa tekstuuri joka on saumaton, sillä jos tekstuuri ei ole saumaton, mallin teksturointi ei näytä yhtenäiseltä. Varsinkin ottaen huomioon että tekstuurikartta on jakanut jo mallimme osiin luoden saumat eri osille, ja saumattomat tekstuurit voivat jakaantua eri osien välille hyvin huonosti luoden epäyhtenäisen lopputuloksen. Loppujen lopuksi työssä päädyttiin valitsemaan neljät eri tekstuurit mallia varten (Kuva 21): Iho tekstuuri kaulaa, rinta-, vatsa- ja käsien lihaksia varten, sarvi tekstuuri sarvia, kynsiä ja kavioita varten, luu tekstuuri hahmon pääkalloa varten, sekä karva tekstuuri loppukehoa varten. Erilliset tekstuuritiedostot käsiteltiin photoshopissa, muokaten valoisuus- ja väriasetuksia saavuttaakseni halutut lopputulokset, esim. tekemällä karvasta vihreää ja sarvitekstuurista valmistettiin kaksi eri versiota. Toisesta sarvitekstuurista tehtiin hieman tummempi jota käytettiin hahmon kavioihin ja kynsiin.



Kuva 21. Valitut mallin tekstuurit jaettuna: 1 Iho, 2 Sarvet, kynnet ja kaviot, 3 pääkallo, 4 keho.

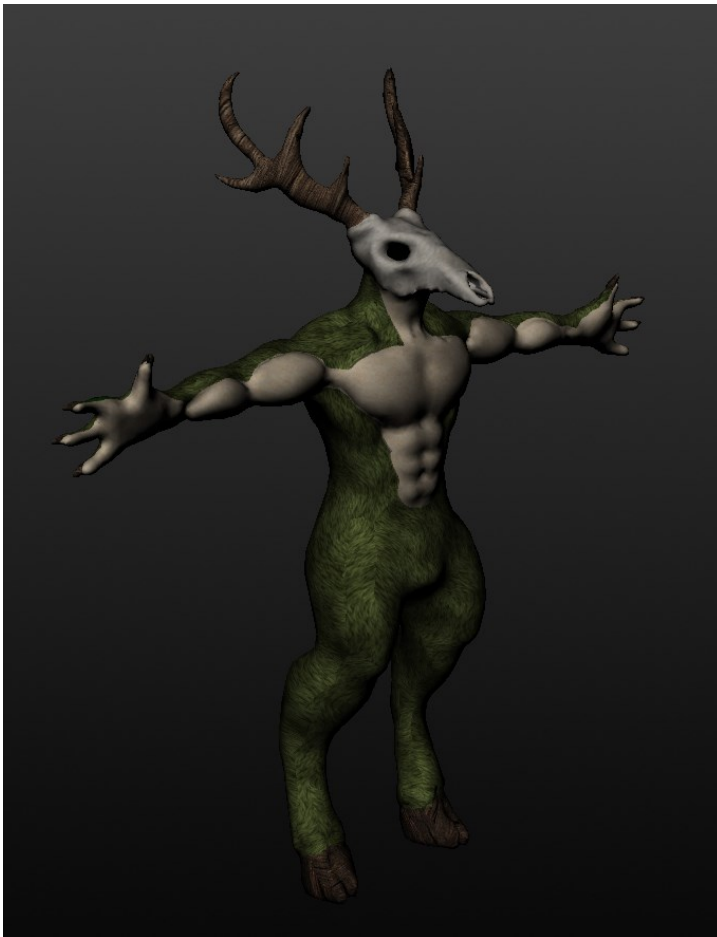
Neliömäisten tekstuuritiedoston applikoiminen tekstuurikartalle toimi rajaamalla photoshopissa halutun alueen tekstuurikartasta. Kun rajausta oli tehty, alueelle tarkoitettu tekstuuuri siirrettiin samaan kuvatiedostoon eri tasolle. Sen jälkeen tekstuuritiedostosta leikattiin rajauksen mukaisesti uusi kuva. Näin saatiin korvattua tekstuurikartasta värikoodattu alue sen teksturoidulla versiolla. Tämä sama prosessi toistettiin jokaisella eri alueella ja tekstuuritiedostolla joka tuotti valmiin tekstuurikartan (Kuva 22).

Tätä prosessia tehdessä on hyvä pitää mielessä että tekstuurit tehdään karttaan oikeassa mittakaavassa hahmoon nähden. Tällä tarkoitetaan esimerkiksi sitä, että karva-tekstuuuri on asetettu tekstuurikarttaan tavalla, että kun tekstuurikartta implementoidaan takaisin mallin päälle, karvat näkyvät oikean kokoisina. Tässä projektissa käytetyt tekstuuritiedostot olivat hyvin isoja verrattuna tekstuurikartan kokoon, joten niiden kokoa piti säädellä halutun tuloksen saavuttamiseksi.



Kuva 22. Texture-map tekstuurien kanssa.

Kun kaikki tekstuurit on saatu asetettua paikoilleen, tiedoston voi tallentaa jonka jälkeen on mahdollista tarkastella miltä malli näyttää juuri tehtyjen tekstuurien kanssa. Sculptriksen paint tilan asetusten kautta uusi päivitetty tekstuurikartta ladataan, jolloin ohjelmisto automaattisesti asettaa sen paikoilleen mallin päälle (Kuva 23). Kun kartta on asetettu takaisin mallin päälle, mallista näkee paremmin mahdolliset ongelmakohdat jota voi lähteä ratkomaan tekstuurikartassamme.

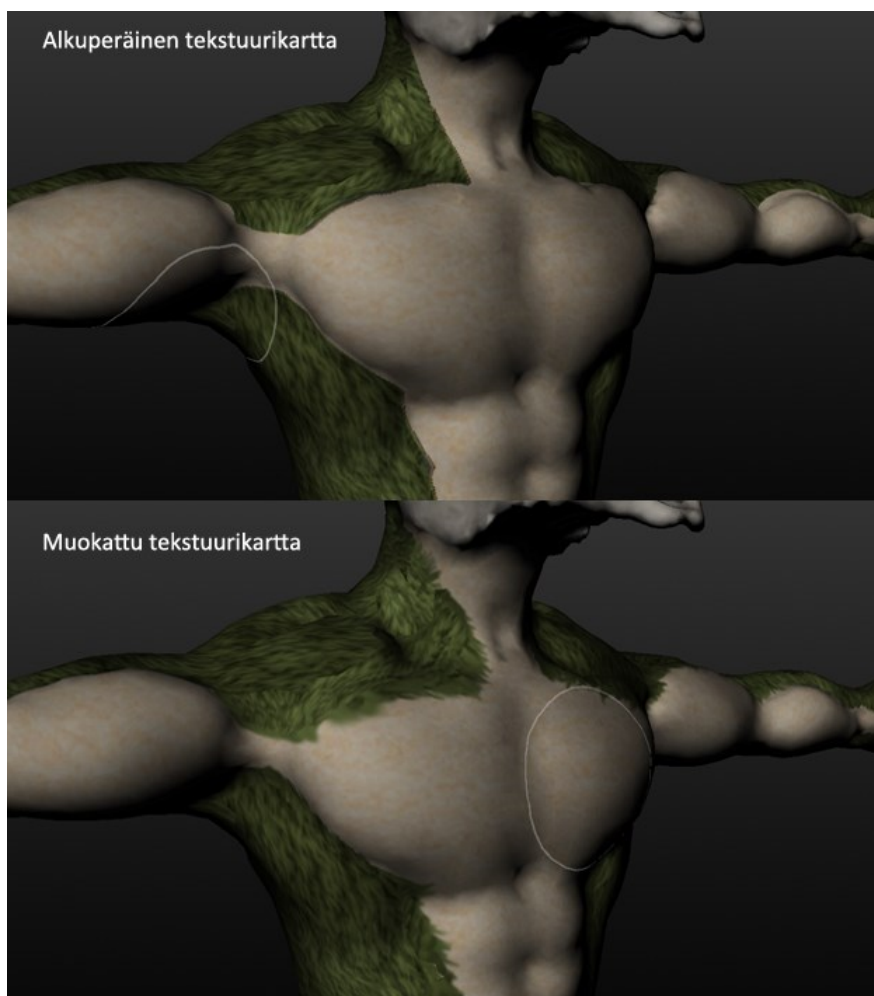


Kuva 23. Tekstuurikartta implementoituna hahmon päälle.

4.3.3 Tekstuurikartan hienosäätö

Heti tekstuurikartan lataamisen jälkeen, näkyviin tulee useita korjattavia kohtia. Koska aikaisemmin seurattiin mallin maalattuja värikoodeja identtisesti tekstuureja laittaessa, tekstuurien väriset erot ja niiden rajakohdat ovat hyvin teräviä ja luonnottoman näköisiä, varsinkin mallin karvatekstuurin ja ihotekstuurin välissä. Tekstuuri kaipaa siis hieman vielä hienosäätöä.

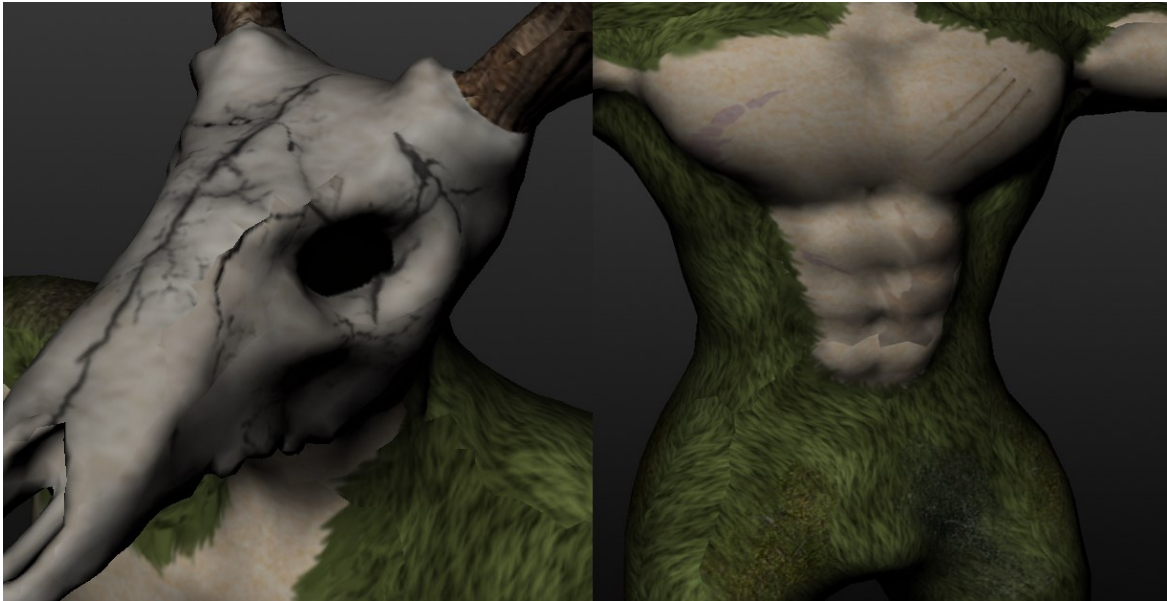
Karva- ja ihotekstuurin rajapintojen korjaaminen hoidetaan suoraan tekstuurikarttaan photoshopin kautta. Tarkoituksena on luoda rajapinnasta luonnollisemman näköisen, joten muokataan kahden tekstuurin välistä rajapintaa tekemällä siitä röpelöisemmän ja pehmeämmän näköisen. Tarkoituksena on simuloida miltä oikeanlainen karvapeite näyttäisi kohdatessa ihon, rintalihasten, kaulan ja käsien kohdalla. Tämä onnistui photoshopin clone stamp ja smudge työkalulla. Karvatekstuuria lisättiin hieman clone stampilla haluttuihin paikkoihin, ja uutta rajapintaa muokattiin tekemällä pieniä yksittäisiä harjauksia smudge työkalun avulla. Tämä sai aikaan efektin että karvatekstuurin rajalta tulisi esiin useita pieniä karvarykelmiä. Uudesta rajasta tehtiin tarkoituksellisesti hieman epätasaisen paikoittain, koska hieman epäsymmetrinen ulosanti mallissa saa aikaan luonnollisemman vaikutelman. Vertailusta voi nähdä kuinka pieni muutos voi saada aikaan paljon luonnollisemman lopputuloksen (Kuva 24).



Kuva 24. Tekstuurikartan hienosäädetyin version vertailua alkuperäiseen versioon.

Tämän lisäksi hahmon tekstuurikarttaa hienosäädettiin lisäämällä muitakin yksityiskoh-
tia, murtumia pääkalloon, hieman arpia ja läikkiä ihon alueille sekä pieniä teksturi- ja
värimuutoksia turkin alueelle (Kuva 25). Yksityiskohdat luotiin liittämällä erinäisiä gra-
fiikoita (arvet, murtumat, erinäiset kasvitekstuurit) tekstuurikarttaan. Ensimmäisenä
merkattiin liitettävien grafiikkojen paikat sculptriksen paint toiminnolla suoraan mallin
päälle, jonka jälkeen merkattu tekstuurikartta avattiin photoshopissa. Merkatun teks-
tuurikartan mukaan grafiikat lisättiin halutuille paikoille.

Grafiikkoja liittäessä on hyvä käyttää tiedostomuotoja jotka tukevat transparency omi-
naisuutta, tarkoittaen että kuvan tausta voi olla täysin tyhjä. Jos tiedosto ei ole tätä
muotoa, kuvasta joutuu poistamaan erikseen taustan, jolloin on mahdollista että gra-
fiikan laatu kärsii. On paljon helpompaa joko luoda tai hyödyntää transparencya tuke-
via tiedostomuotoja. Tässä tapauksessa grafiikat olivat png tiedostomuotoa, joten ne
pystyttiin liittämään suoraan tekstuurikarttaan. Grafiikkojen liittämisen jälkeen, niitä
käsiteltiin hieman erivärisiksi ja säädettiin sekä kontrasti että läpinäkyvyys asetuksia
jotta ne sulautuisivat paremmin tekstuurikartan muuhun ympäristöön.



Kuva 25. Lisättyjä yksityiskohtia mallin tekstuuriin.

4.4 Tuominen pelimoottoriin

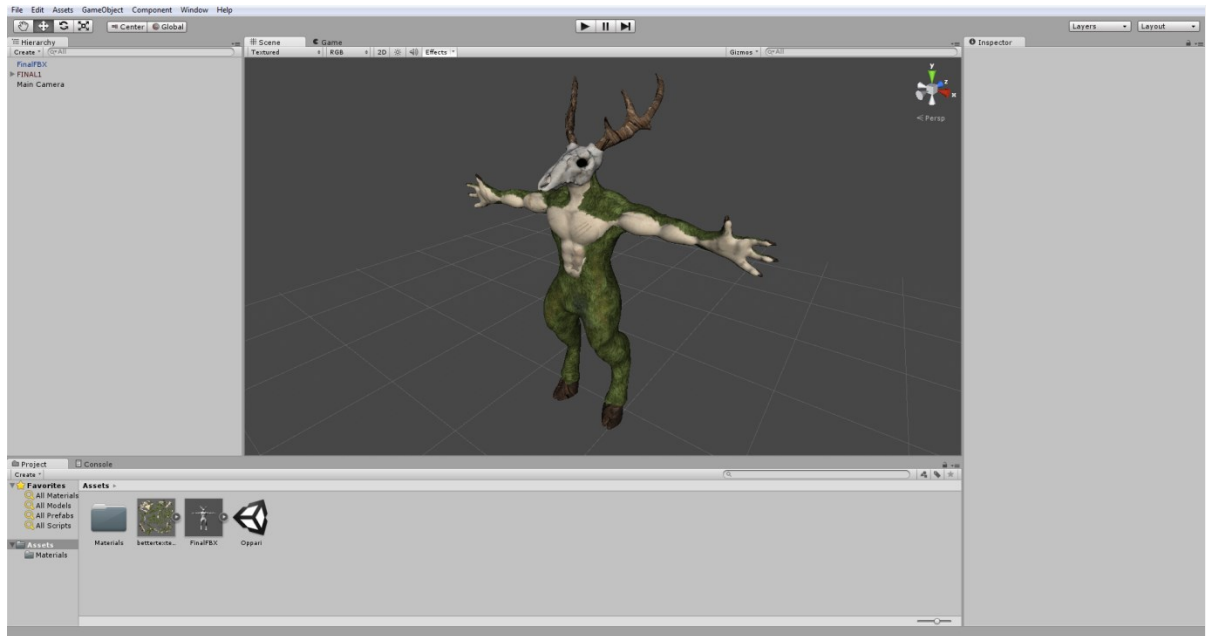
Seuraavana ja viimeisenä työvaiheena projektissa on mallin tuominen pelimoottoriin. Tämä prosessi on suhteellisen yksinkertainen, kuten kuvassa 26 on illustroitu.



Kuva 26. Projektin työnkulku, siirtyminen pelimoottori vaiheeseen.

Kun tekstuurit on hoidettu haluttuun pisteeseen, seuraavana askeleena työssä on siirtää malli pelimoottoriin. Sculptriksen omat tiedostot ovat tiedostomuotoa sc1, jota suurin osa muista mallinnusohjelmista tai pelimoottoreista ei valitettavasti tue, joten tiedosto exportataan ulos obj tiedostomuotona. Tarkoituksena on kuitenkin konvertoida juuri saatu obj tiedostomme joksikin muuksi, ja se hoidetaan toisessa ohjelmistossa, sillä sculptris pystyy exporttaamaan tiedostoja ulos vain obj tiedostomuotona. Tässä projektissa malli muutetaan fbx tiedostoksi, sen yhteensopivuudesta pelimoottorin kanssa johtuen.

Kun tiedosto on muunnettu haluttuun tiedostomuotoon, malli siirretään unityyn ohjelmiston import toiminnon kautta, valitaan tehty mallimme jolloin malli ilmestyy pelimoottorin käyttöliittymään. Tekstuuri implementointi tapahtuu niinkin yksinkertaisesti, kun vetämällä luodun tekstuurikartan hahmon päälle, unity automaattisesti asettaa tekstuurikartan oikein hahmon päälle (Kuva 27). Tämän jälkeen hahmo, sekä projekti on valmis. Tässä vaiheessa 3D-artistin työprosessi on finalisoitu ja mallin kanssa työn tekoa jatkaisi muut pelinkehittäjät pelimoottorin sisällä.



Kuva 27. Teksturoitu malli pelimoottorissa.

5 POHDINTAA

Pohdinnan tarkoituksena on tutkia käytyä projektin työprosessia (Kuva 28), projektille asetettuja tavoitteita ja miten ne täyttyivät työn eri osa-alueitten, sekä koko projektin suhteen.



Kuva 28. Projektin työkulku asteittain.

Sculptaus itsessään ei pidä sisällään hirveästi rajoitteita loppuprojektin kannalla teknisesti, joten varsinaisen mallin luominen oli hyvin vapaamuotoista. Suurimmat tekniset rajoitteet tässä työssä tulevat re-topology käsittelyn ja teksturoinnin kohdalla, ja niissä tehdyt virheet tai huolimattomasti tehty jälki voi aiheuttaa suuriakin ongelmia myöhemmässä osassa työtä, ja jopa tehdä mallista käyttökelvottoman kun sitä koetetaan hyödyntää varsinaisessa käyttötarkoituksessaan. Pelkästään sculptausta ajatellen, projekti onnistui loistavasti. Sculptriksella pystyi luomaan hyvin helposti ja vapaasti äärimmäisen fotorealistisen lopputuloksen, ja ohjelmisto ei asettanut rajoitteita sculptaus työn etenemiselle, joten ainoina rajoitteina tässä osassa projektia olivat mallintajan omat kyvyt sekä mahdolliset aikarajoitteet. Verrattuna klassisempiin mallinnustapoihin jota muut ohjelmistot hyödyntävät, vastaavanlaisen lopputuloksen saavuttaminen on lähes mahdotonta. Sinänsä on hieman hyödytöntä verrata keskenään digitaalista sculptausta klassisempiin ja vanhemman luokan mallinnustekniikoihin, sillä nämä kaksi ovat täysin eri lähestymistapoja samaan työkuvaan, ja ne ovat tarkoitettu saavuttamaan erilaisia lopputuloksia, täysin eri tavoilla.

Re-topologyn kanssa, projektissa asetettu polygonimäärän tavoite saavutettiin odotetusti. Polygonimäärä vähentyi yli kymmenkertaisesti, ja malli säilytti visuaalisen ulkomuotonsa ja tarpeelliset yksityiskohdat ja lopputuloksena malli sisälsi noin kaksikymmentätuhatta polygonia. Vaikka sculptris on suunnattu aloittelijoille, ja sen re-topology työkalutkin ovat helpommat ymmärtää ja käyttää kuin muut vastaavien ohjelmistojen vastakappaleet, niitten vapaamuotoinen käyttötapo voi koitua hieman ongelmallisiksi projektin kannalta. Re-topology käsittely muissa ohjelmistoissa voidaan hoitaa hyvin erilaisilla tavoilla, mutta päämääräisesti tämä työvaihe on hyvin vakaasti linkitetty mallin UV-kartoituksen kanssa. Riippuen miten re-topology hoidetaan mallissa, lopputulos vaikuttaa dramaattisesti mallin UV-kartoituksen perusteella luotuun tekstuurikarttaan, ja koska sculptriksen re-topology käsittely onkin hyvin vapaamuotoinen, näiden kah-

den osa-alueen (re-topology ja UV-kartoitus) välillä ei synny samanlaista jatkuvuutta kuin muissa vastaavissa ohjelmistoissa. Tästä syystä hyvän ja helppokäyttöisen tekstuurikartan luominen sculptriksessa vaatii joko paljon tietotaitoa tai vaihtoehtoisesti jatkuvia yrityksen ja erehdyksen kautta tapahtuvia korjauksia. Tämä ei tietenkään tarkoita sitä että sculptriksella tekstuurikarttojen luonti olisi mahdotonta, mutta jos vastaavanlaista projektia olisi työstämässä, olisi todennäköisesti kätevämpää käyttää muita toimintatapoja tai ohjelmistoja tähän tarkoitukseen. Tämän projektin, ja sen jatkuvuuden kannalta kaikkiin mallinnukseen liittyviin työvaiheisiin (sculptaus ja re-topology) käytettiin sculptrista.

Teksturointi itsessään onnistui tyydyttävästi. Mallin ”eloon” tuomisessa on hyvin olennaista että teksturointi hoidetaan järkevästi jotta pelihahmon visuaalinen ulosanti on mahdollisimman hieno. Teksturoinnin työprosessin kannalta on siis olennaista että tekstuurit valitaan ja implementoidaan tarkasti mallin päälle, jossa myös UV-kartoitus on isossa osassa. Oikein suoritettu UV-kartoitus, mallin rajapintojen ja tekstuurikartan saumojen sijoittaminen malliin ovat kaikki osa-alueita jotka voivat tuntua hyvin ohimeneviltä työvaiheilta mutta ovat lopputuloksen ja koko teksturoinnin kannalta hyvin olennaisia. Niin kuin aikaisemmin tässä projektissa mainittiin, sculptriksen omat re-topology työkalut ja niitten käyttö tuottivat tekstuurikartan jota pystyi käyttämään projektissa, mutta kyseisen tekstuurikartan hyödyntäminen oli loppujen lopuksi hyvin yksityiskohtaista työtä ja vaati paljon hiomista ja säätämistä tekstuureja muokatessa. Tekstuurien luonti ja implementointi onnistuneesti oli työprosessina täysin samanlainen riippuen käytettävästä mallista tai kartasta, mutta jälkeinpäin ajatellen, teksturointi itsessään olisi onnistunut huomattavasti helpommin ja nopeammin jos re-topology olisi hoidettu eri tavalla, ja mallin UV-kartoitus olisi hoidettu manuaalisesti, sculptriksen oman automaattisen kartoittamisen sijasta.

Mallin tuominen pelimoottoriin oli pohjimmiltaan vain projektin finalisointia, ja ei itsessään vaatinut paljon. Kyseessä oli enää vain teknistä toimintaa jonka kanssa ei oikeastaan voinut epäonnistua, ja malli loppujen lopuksi saatiin importoitua onnistuneesti pelimoottoriin tekstuurien kanssa. Jos tämän projektin tutkimista jatkettaisiin vielä pelimoottorin sisällä, projektissa tulisi esiin useita seikkoja jotka ovat hyvin vahvasti linkittyneitä tämän projektin osa-alueisiin. Kun mallia aletaan työstää esimerkiksi animointia varten, on olennaista huomioida mahdollinen mallin meshin ja luotujen tekstuurien venyminen mallia liikuttaessa, aiheuttaen tekstuurien ja mallin olomuodon vääristymistä. Vastaavanlaiset seikat ja niistä huolehtiminen olisi yleisesti 3D-artistin työnkuvassa, mutta tämän projektin suoraviivaisuuden ja tarkoitetun aihealueen myötä vastaavat työprosessit on ajateltu tapahtuvan projektin rajojen ulkopuolella, eli pelimoottorikäsittelyssä.

Kaiken kaikkiaan, projektin tavoitteet täyttyivät päämääräisesti ja asiat joita oli tarkoitus saavuttaa, saavutettiin. Pieniä huomiokohtia nousi, joissa projektin aikana sekä jälkeen joutui pohtimaan olisiko tämä kannattavaa tehdä erilailla tai eri ohjelmalla, joista tuli hyviä kehitysideoita ja parannettavaa seuraavalle vastaavanlaiselle projektille.

LÄHTEET

- Autodesk (2014) Introduction to UV mapping. Haettu 5.2.2018 osoitteesta <https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/Maya/files/UV-mapping-overview-Introduction-to-UV-mapping-htm.html>
- Baker, M. (2016) How do Game Engines work? Haettu 3.1.2018 osoitteesta <https://interestingengineering.com/how-game-engines-work>
- Brodin, J. (2013) How Unity3D became a game-development beast. Haettu 22.1.2018 osoitteesta <http://insights.dice.com/2013/06/03/how-unity3d-become-a-game-development-beast/>
- Fine, R. (2017) UnityScripts long ride off into the sunset. Haettu 22.1.2018 osoitteesta <https://blogs.unity3d.com/2017/08/11/unityscripts-long-ride-off-into-the-sunset/>
- Franklin, C. (2000) How 3D graphics work. Haettu 22.1.2018 osoitteesta <https://computer.howstuffworks.com/3dgraphics1.htm>
- Ghosh, D. (2014) Theory of 3D modeling. Haettu 5.1.2018 osoitteesta <http://www.sketchup-ur-space.com/2014/aug/theory-of-3D-modeling.html>
- Martin, J. (2016) How to re-topologize 3D scans into low poly game assets. Haettu 4.1.2018 osoitteesta <https://blog.sketchfab.com/retopologise-3d-scans-low-poly-game-assets/>
- O'Rourke, M. (2012) Intro to polygonal modeling. Haettu 22.1.2018 osoitteesta <http://3dtutorials.michaelorourke.com/tutorials/Modeling/Basics/PolyModIntro12.pdf>
- Pluralsight (2014) What's the difference? A comparison of modeling for games and modeling for movies. Haettu 6.2.2018 osoitteesta <https://www.pluralsight.com/blog/film-games/whats-the-difference-a-comparison-of-modeling-for-games-and-modeling-for-movies>
- Stylianou, O., Vlachaki, N. (2017) What is digital sculpting? Haettu 22.1.2018 osoitteesta <https://www.themightygorgonion.com/2017/07/30/what-is-digital-sculpture/>
- Unity Technologies-a (2017) Unity User Manual , Shaders. Haettu 7.2.2018 osoitteesta <https://docs.unity3d.com/Manual/Shaders.html>
- Unity Technologies-b (2017) Unity User Manual , Animation overview Haettu 7.2.2018 osoitteesta <https://docs.unity3d.com/2017.3/Documentation/Manual/AnimationOverview.html>

Unity Technologies-c (2017) Unity User Manual, Networking overview. Haettu 7.2.2018 osoitteesta

<https://docs.unity3d.com/Manual/UNetOverview.html>

Ward, J. (2008). What is a Game Engine? Haettu 3.1.2018 osoitteesta

https://www.gamecareerguide.com/features/529/what_is_a_game_.php

Wikipedia-a.(n.d). Sculptris. Haettu 5.1.2018 osoitteesta

<https://en.wikipedia.org/wiki/Sculptris>

Wikipedia-b. (n.d). Digital sculpting. Haettu 5.1.2018 osoitteesta

https://en.wikipedia.org/wiki/Digital_sculpting

Wikipedia-c.(n.d). Voxel. Haettu 7.1.2018 osoitteesta

<https://en.wikipedia.org/wiki/Voxel>

Wikipedia-d.(n.d). Game engine Haettu 9.1.2018 osoitteesta

https://en.wikipedia.org/wiki/Game_engine

Wikipedia-e.(n.d). UV mapping Haettu 10.1.2018 osoitteesta

https://en.wikipedia.org/wiki/UV_mapping

Wikipedia-f.(n.d). Unity (game-engine) Haettu 7.2.2018 osoitteesta

[https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))

Wodehouse, C. (2016) 3D Modeling: A Look at the Science and Art Behind It. Haettu 5.1.2018 osoitteesta

<https://www.upwork.com/hiring/design/3d-modeling-the-science-and-art-behind-it/>